Leveraging Spatial Information in Smart Grids using STGCN for Short-Term Load Forecasting

CHUNG MING CHEUNG, University of Southern California, USA
SANMUKH RAO KUPPANNAGARI, University of Southern California, USA
RAJGOPAL KANNAN, US ARL-West, USA
VIKTOR K. PRASANNA, University of Southern California, USA

The problem of predicting the behaviour of energy consumers (loads) in the next few intervals — Short-Term Load Forecasting (STLF) is critical to the success of several grid operations. Prediction at lower aggregation levels is difficult due to the high volatility of the data. Smart grid operations, and in turn any data generated as a result of them, exhibit high spatial correlations imposed due to the topology of the power distribution network as well as other latent factors such as similarity in neighborhood, socio-economic status, etc. While temporal information is usually leveraged in neural network structures like Recurrent or Convolutional Layers, the use of spatial information in load forecasting has not been explored.

In this paper, we develop a Spatial-Temporal Graph Convolutional Network (STGCN) model for the problem of Short-Term Load Forecasting in Smart Grids. STGCNs specialize in capturing both spatial and temporal correlations in the data to obtain more accurate predictions. We also show that our model, by capturing both spatial and temporal correlations, is more robust to missing data than state-of-the-art prediction models. We perform detailed evaluation on a dataset based in Iowa, US with real power at a low aggregation level (5~10 customers per datapoint) and show that our model predicts 3 hours ahead real load consumption with a Mean Absolute Error of 7.54% less than the best performing baseline model, and as much as 38.72% less in Root Mean Squared Error (RMSE) if the data has missing entries.

CCS Concepts: • **Hardware → Smart grid**; • **Computing methodologies → Neural networks**.

Additional Key Words and Phrases: neural networks, short term load forecasting, smart grids, spatial-temporal data

ACM Reference Format:
Chung Ming Cheung, Sanmukh Rao Kuppannagari, Rajgopal Kannan, and Viktor K. Prasanna. 2021. Leveraging Spatial Information in Smart Grids using STGCN for Short-Term Load Forecasting. 1, 1 (May 2021), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Proliferation of AMI meters which allow bi-directional communication and control has enabled utilities to undertake dynamic decision making programs such as Demand Response to optimize grid operations [1]. The problem of predicting the behaviour of energy consumers (loads) in the next few intervals — Short-Term Load Forecasting (STLF) is critical to the success of such programs [11].

Authors' addresses: Chung Ming Cheung, chungmin@usc.edu, University of Southern California, Los Angeles, USA, 3740 McClintock Ave; Sanmukh Rao Kuppannagari, kupanna@usc.edu, University of Southern California, Los Angeles, USA, 3740 McClintock Ave; Rajgopal Kannan, rajgopak@usc.edu, US ARL-West, Los Angeles, USA; Viktor K. Prasanna, prasanna@usc.edu, University of Southern California, Los Angeles, USA, 3740 McClintock Ave.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
Manuscript submitted to ACM

Typically, STLF refers to a prediction window of 1 hour to a day or even a week, while medium-term or long-term load forecasting refer to prediction windows of even longer terms [12]. Moreover, the main focus of STLF is to inform dispatch related decisions such as DER/generator scheduling, demand/solar curtailment, etc. as opposed to medium or long term which are used for planning of grid operations or procurement of energy [4]. Accuracy of STLF is critical for dispatch related decision making to avoid disturbances such as frequency fluctuations, voltage mismatch, etc. due to supply demand mismatch.

Due to the high impact of the problem, a plethora of works have focused on STLF and researchers are still exploring novel ways to further improve the accuracy. The difficulty of STLF is proportional to the aggregation level of the load data. Aggregation level refers to the number of customers whose data are aggregated to consitute a single data point. Research has shown that it is difficult to obtain accurate results for STLF at low aggregation levels [13]. This is because individual customers have very volatile consumption patterns with many spikes in energy usage due to sharp changes caused by turning on or off large electric appliances. Aggregating a few customers is insufficient to smooth the curve and reduce volatility. While STLF on high aggregation level data provides better prediction accuracy, it loses neighborhood level or individual customer level information which is needed for efficient implementation of programs such as demand response.

Data-driven deep neural network models have obtained state-of-the-art prediction accuracy for the problem of STLF. However, these models train on individual time series data and thus are only able to capture long term complex temporal correlations. Smart grid operations, and in turn any data generated as a result of them, exhibit high spatial correlations imposed due to the topology of the power distribution network as well as other latent factors such as similarilty in neighborhood, socio-economic status, etc. Modeling of spatial correlations, in addition to the temporal correlations can further improve the accuracy of STLF. Robustness of STLF is another key research problem. Data collected from smart meters is prone to missing values due to reasons such as unexpected device power off, communication failure, etc. [14, 16]. Missing values lead to reduced accuracy in STLF. Thus, techniques to improve robustness of prediction models are needed.

In this paper, we propose a Spatial-Temporal Graph Convolutional Network (STGCN) model for the problem of Short-Term Load Prediction in Smart Grids. STGCNs are a popular class of GCNs with the ability to learn both temporal correlations and spatial correlations in data. In our STGCN, we use the power distribution network topology as the underlying graph to model spatial correlations. We believe that the network is effective for STLF because the consumption patterns of customers in the same neighborhood can be correlated in several ways. These customers experience common external factors that can affect energy usage, e.g. temperature, weather conditions, also they may be from similar demographics which may lead to similar energy usage like a neighborhood consisting of middle-class income families. In addition, since STGCN can pass information between neighboring customers, it can be more robust to anomalies in data. For example, if a customer's data is missing for some time due to error in measurements, conventional models will be greatly affected by these wrong measurements and prediction accuracy would plummet. On the other hand, an STGCN can make use of neighbor data to make an estimate on the consumption of the customer data during the missing data period.

Our proposed STGCN model is able to obtain state-of-the-art prediction accuracy for STLF. We also show that our model, by capturing both spatial and temporal correlations is highly robust to missing data. Specifically, the contributions of this paper are as follows:

- We develop a STGCN model to capture both spatial and temporal correlations for the problem of STLF.

- We develop a methodology to evaluate the performance of the model under different types of missing data prevalent in smart grids.
- We perform detailed evaluation on a dataset based in Iowa, US which consists of power data at a low aggregation level (5∼10 customers per datapoint) and show that our model is better in real load prediction than other baselines.
- Our evaluations also show that STGCN models exhibit significantly better performance than the baseline model if the dataset has missing values.

2 RELATED WORKS

Proliferation of AMI meters in smart grids has enabled data collection at finer granularity (few minutes to an hour) [1]. This has propelled the research in the development of data-driven solutions for several problems in smart grids [3]. One such problem is the prediction of customer consumption for future horizons, also known as load forecasting.

The problem of load forecasting has received widespread attention and several methods have been developed for accurate prediction of future load demand [20]. Traditionally, statistical methods like moving average or ARIMA [15] have been the most popular methods for load forecasting due to their low complexity and low computational overhead. However, as computational power of computers increased, data-driven models have surpassed the popularity of statistical models as they can represent complex non-linear relationships. Models like Support Vector Regression and Random Forest Regression have been widely successful in many applications including time series analysis and forecasting [6].

Recently, state-of-the-art deep neural networks such as Long-Short Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) have been applied to the problem of STLF. These networks are capable of representing even more complex models than traditional ML and can deliver state-of-the-art performance. For example, [10] uses LSTMs to learn representations for sequential data. In addition to the historical loads, time related features like the day of the week, the time interval of the day, and whether the day is a holiday are used. [18] uses CNNs to learn correlations of data within a certain temporal locality and make predictions, this paper also uses k-Means to segment the dataset into subsets of data point clusters to construct an ensemble of CNN models. While both kinds of networks are powerful, neither is able to consider spatial information in smart grids as inputs.

3 PROBLEM DEFINITION

3.1 STLF Definition

Given the time series of load consumption of customers in a smart grid, the problem of short-term load forecasting is to predict the load consumption values for each customer for the next few intervals. We are also given the topology of the power distribution network.

Formally, the problem of load forecasting that we consider in this paper is defined as follows: Let $\mathbf{x}_{1:T} = \{x_1, x_2, x_3, ..., x_T\}$ denote the input time series, where the data point at each time interval t is represented by x_t and represents the vector of load consumption values of the customers in the grid. Let x_t^i denote the data point for customer i. We are also given a window size W that denotes the length of previous intervals to use for prediction and a future horizon size H that denotes the number of future intervals to predict. The problem of STLF is to find a prediction function f such that:

$$x_{t+W+1:t+W+H} = f(x_{t:t+W}) \tag{1}$$

The function f can be any regression model or neural network architecture.

Now let us assume that the power grid distribution network topology graph g is provided to us with a customer i represented as node i on the graph. The edges in the graph represent the feeder lines between two customers. In this case, we extend the problem of STLF to consider both spatial and temporal correlations and learn a function f' such that:

$$x_{t+W+1:t+W+H} = f'(x_{t:t+W}, g) \tag{2}$$

3.2 STLF with Missing Data

In real-world deployment scenarios, the data obtained from the metering infrastructure is prone to missing or corrupted values [14]. The missing or corrupted values are often labelled with 0s or NaN. STLF on such data may lead to very large errors in predictions which is undesirable.

We hypothesize that by leveraging spatial information, STLF models can be robust to missing data. To test the hypothesis, we evaluate the robustness of our STLF models by testing their accuracy via a modified testing methodology (Section 5.1.2). We remove some data points from our test dataset X_{te} using missing data patterns that occur widely in the field and reflect several typical real-world missing data scenarios [14]. Let $\mathbf{x}'_{te}$ denote this modified test dataset.

In our modified testing methodology, the testing dataset predictions $\hat{\mathbf{x}}$ are acquired using $\mathbf{x}'_{te}$.

$$\hat{\mathbf{x}} = f'(\mathbf{x}'_{te}, g), \tag{3}$$

while the error is evaluated between the predicted values and the unmodified testing dataset.

$$Error = E(\hat{\mathbf{x}}, \mathbf{x}_{te}) \tag{4}$$

4 METHODOLOGY

4.1 Graph Convolutional Network (GCN)

Graph Convolutional Network (GCN) was first developed by Kipf and Welling [9] for the classification of nodes in a graph network. It is a variant of CNN that performs convolution directly on graphs, this convolution allows each node to aggregate the input representations of neighboring nodes to form a new representation. This allows information exchange between neighboring nodes to learn spatial dependencies among these nodes.

The inputs to a GCN layer is a matrix of N data points, each with D_i dimensions at layer i. Each layer of GCN performs a matrix multiplication on the inputs and the adjacency matix. To avoid changing the scale of the inputs, the adjacency matrix is normalized by symmetric normalization using the diagonal node degree matrix. Self-loops are also added for each node to the adjacency matrix so that the features of each node itself are also considered when computing the outputs for each layer. Mathematically, a GCN layer can be written as

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \tag{5}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times D_l}$ denotes the l^{th} layer of the network with its weight parameters denoted as $\mathbf{W}^{(l)}$. $\mathbf{A}$ denotes the adjacency matrix of the graph g, and $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is defined as the adjacency matrix with added self-connections, where $\mathbf{I}_N$ is the identity matrix. $\tilde{\mathbf{D}}$ is the degree matrix calculated by the corresponding modified adjacency matrix $\tilde{\mathbf{A}}$ by $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.

Efficient methods for implementing GCNs are discussed in [9].

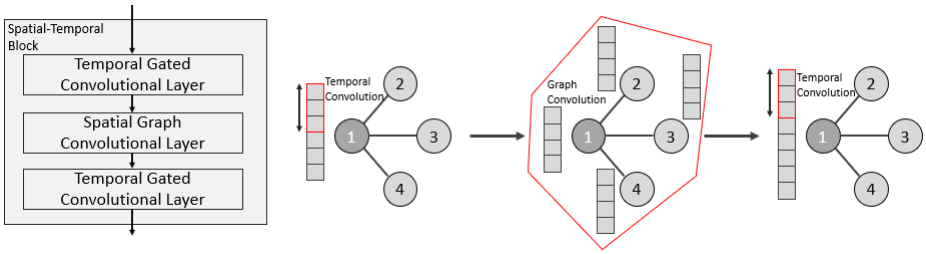

Fig. 1. Architecture of a Spatial-Temporal Convolutional Block (left) and Illustration of Convolutions (right)

4.2 Spatial-Temporal Graph Convolutional Network (STGCN)

While a GCN layer only learns spatial correlations, Spatial-Temporal Graph Convolutional Network (STGCN) forms spatial-temporal convolutional blocks which consists of gated convolutional layers to perform convolution in the temporal dimension and GCN layers to discover spatial correlations [19].

A spatial-temporal convolutional block is structured as a gated convolutional layer in the temporal dimension followed by a GCN layer and another gated convolutional layer. The structure of this convolutional block is illustrated in Figure 1. Denoting f_g as the graph convolution operation as described in Section 4.1, f_{c1} and f_{c2} as gated convolutional layers, and x_t as the input at time interval t, the spatial-temporal convolutional block function can be written mathematically as

$$x_t = f_{c2}(f_g(f_{c1}(x_t))) \tag{6}$$

Overall, the STGCN layers takes from each node, a time series input of the historical load sequences of that node, a temporal convolution is performed along the axis of the input of each node, the output is then passed to a graph convolution layer to perform convolution across neighbor nodes inputs, a second temporal convolution is then performed. This is illustrated in Figure 1. A bottleneck strategy is applied within the block, meaning the output feature size of the GCN layer is smaller than the two temporal layers, forcing the network to learn a meaningful representation of the inputs that considers both spatial and temporal dependencies. A fully connected output layer is applied at the end of the model to translate the learned representation from convolutional blocks to the targeted prediction values.

In this paper, the STGCN model we used is similar to STGCNs used in other applications like traffic flow prediction [19]. In the next subsection, we elaborate the steps we have taken to apply STGCNs to power grids load forecasting.

4.3 Application of STGCN to Power grids

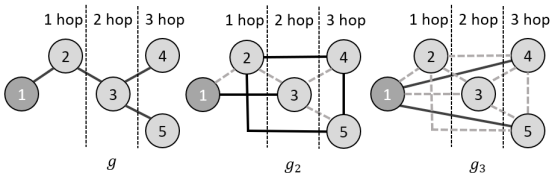

Fig. 2. Modification on graph by appending edges to multihop neighbors

To apply STGCNs for power grid networks, we need to define the grid as a graph. The nodes of the graph represents loads of customers, and edges connect nodes according to how transformers and loads are connected in the smart grid

topology. However, edges do not represent a uniform distance between the nodes, so customers that are connected on the topology may actually be very distant, while nodes two hops away (meaning it is reached by going through two edges) can actually have a small distance between them.

To include this information into the GCN modelling without making the GCN layer more complex, we choose to modify the adjacency matrix to add edges for nodes that are more than one hop away. We denote g as the original graph which is the conversion of the topology to a graph, g^i is the graph where all nodes that are i-hops away in g have edges between them in g^i. The number of hops to include is a hyperparameter that needs to be tuned. This process is illustrated with an example in Figure 2, where the number of hops needed to reach a node from node 1 is labelled on top of the graph. For g_2, the existing edges in g is colored as gray dashed lines to emphasize the appended edges with solid black lines, an edge is added from node 1 to node 3 because they are 2 hops apart, the same is done for other nodes. For g_3, similarly, two edges are appended compared to g_2, one between node 1 and node 4, another one between node 1 and node 5, because these nodes are 3 hops apart. We decide the ideal number of layers of GCN and number of hops to use by evaluation on the validation dataset.

5 EXPERIMENTS

5.1 Experimental Setup

In our experiments, we compare the performance of the proposed STGCN against several baselines. A summary of the algorithms is shown in Table 1.

Table 1. List of Algorithms used

Algorithm	Description
SGD + Nystroem kernel [17] approximate	A simple but effective kernelized regression model solved by stochastic gradient descent.
Convolutional Neural Network	Neural networks with convolution layers used to discover temporal correlations for STLF [8].
LSTM	Univariate LSTM network [10] that takes only the load as input.
LSTM Multivariate	LSTM network with additional features related to time and customers.
STGCN	Neural network that specializes in discovering both spatial and temporal correlations.
STGCN Multi-hop	STGCN with the input graph including nodes more than one hop away as neighbors.

STLF problems are usually defined as load forecasting problems that predict up to a day in the future [2]. We evaluate prediction models for predicting 1, 3, and 6 hours into the future to understand the performance of the the proposed models under various scenarios.

We perform two different experiments to evaluate the benefits of using STGCNs over existing state-of-the-art algorithms for load prediction in realistic settings:

5.1.1 Real Load Prediction. We first consider the standard STLF problem. For each algorithm, a predictor tries to predict the real consumption load of the next 6 hours in time given a subsequence of historical real consumption load of a node or all neighboring nodes in the case of STGCN.

Inputs: Historical time series of real consumption load, power grid topology

Outputs: The real consumption load prediction in the next 6 hours at hour intervals

Evaluation Metrics: MAE, MAPE and RMSE between predicted real load and actual real load

5.1.2 Missing data experiments. The objective of the second experiment is to test the robustness of the algorithms when anomalies occur in the input data. A common anomaly is missing values in the datasets due to failures on the data collection side, missing or corrupt values are usually caused by failures in communication, or unexpected device power offs. This causes a period of "NaN" or zero values in the time series that are either for a single interval or up to half a day, and occasionally up to a full day. In this experiment, we modify the testing dataset such that some data values are missing, and the prediction model needs to perform forecasting with the modified dataset. This experiment tests the robustness of the models for their ability to perform in real life scenarios.

To obtain dataset with missing values that reflect real life situations, we use the results of the study performed in [14] summarized in Table 2 which analyzes the various patterns exhibited by missing data and their statistics. According to the study, ∼47% of missing data constitutes of single missing entries while ∼51% of missing data constitutes of a block of missing data with period up to half a day. Therefore, we introduce either single entries or block of sizes less than half a day as missing data as they constitute the majority of missing data in real life scenarios.

We consider two methods to represent the missing values in the dataset. The first method consists of setting the missing values to 0, because unlike "NaN" values, zeros can still be processed by prediction models. The second method is to perform linear interpolation of missing values prior to using them for STLF. Linear interpolation is a popular preprocessing step to clean the data of any missing values before feeding into a Machine Learning model [7]. Thus, we also consider this step to reflect real world deployment scenarios.

Now, to generate the missing data, we start with picking the nodes which will have missing data. We denote this set as V_m i.e., the set of nodes with missing data. We consider two methods for the choice of these nodes. The first method is to use *random selection*, where the choice of nodes is completely random. This represent failures (intermittent or prolonged) of meters at the nodes. The second method is to use *spatial locality selection* that selects clusters of nodes which are spatially close to each other. This represents failures such as communication blackouts which impact entire neighborhoods. For the second method, we start with a randomly selected set of nodes. For each node in V_m, we explore all the neighboring nodes (if not already explored) and decide if they should be included in the set with probability inversely proportional to the distance between the two nodes. This is repeated until no more nodes are chosen in an iteration. This method creates clusters of nodes which are close to each other spatially as per the power grid topology.

Table 2. Missing value periods statistics [14]

Category	Period	No. of Intervals	Percentage
1	single missing interval	1	47.14%
2	up to half day	(1,12]	51.26%
3	up to one day	(12,24]	1.05%
4	up to one week	(24,192]	0.53%
5	more than one week	(192	0.02%

For each node or cluster in V_m, the length of the each period of missing values have to be decided. In our experiments, we either use fixed length, or length determined by real life statistics. For fixed length, we assume that all periods of missing values would have the same predetermined length for all nodes, this allows us to analyze how load forecasting algorithm performance would change according to how long the missing data periods are. For statistics, we follow the study done by [14] on a real life dataset with $72k$ missing periods, on the distribution of lengths of missing value periods.

These statistics are shown in Table 2. The data is then generated by first deciding for each missing value period, the length category it belongs to, then randomly generating the length of the period according to the determined category. The results from using this method would be able to better reflect the performance of the algorithms in real life.

We perform four different experiment setups as described in Table 3. For the number of customers with missing data (k), we set $k = 10$ for random selection, and set 5 as initial number for spatial locality selection, this gives $k = 15$ on average. We also vary the length of each missing data period $l = \{6, 12, 24\}$ intervals, and the number of missing data periods are set to $\{40, 20, 10\}$ respectively with respect to l. This creates a total of 10 days of missing data regardless of the length of each missing data period, which means about 13.6% of the testing data is replaced for the chosen customers. In addition, we also generate a dataset labelled "stat" which contains generated missing data where the missing data period is distributed according to a real-life dataset distribution as described in Section 5.1.2. To evaluate the output of the predictor, the predicted values are compared to the original measured values of the consumption before the modification.

Table 3. Setups for Missing data experiments

Setup No.	Missing Data Values	Selection
1	Zero Values	Random
2	Interpolated	Random
3	Zero	Spatial Locality
4	Interpolated	Spatial Locality

Inputs: Historical time series of real consumption load with missing data, Power grid topology

Outputs: The real consumption load prediction in the next 6 hours at hour intervals

Evaluation Metrics: MAE, MAPE and RMSE between predicted load and actual load before modification on customers with missing data only

5.2 Evaluation Metrics

In our experiments, we use the following widely used error functions to evaluate the algorithm performances: Mean Absolute Error (MAE), Mean Average Percentage Error (MAPE) and Root Mean Squared Error (RMSE).

MAE is the $\ell 1$-norm of the absolute difference between the actual values and the predicted values. Using MAE takes data points of larger values with greater importance, as they are more likely to have a larger error. MAPE is the mean of the percentage difference between the actual values and the predictions. So, unlike MAE, it takes the error of each data point of equal significance regardless of the amplitude of their values. RMSE is the square root of the $\ell 2$-norm of the difference between the actual values and the predicted values.

$$MAE = ||\frac{\mathbf{x} - \hat{\mathbf{x}}}{N}||_{\ell 1} \tag{7}$$

$$MAPE = \frac{1}{N}||\frac{\mathbf{x} - \hat{\mathbf{x}}}{\mathbf{x}}||_{\ell 1} \tag{8}$$

$$RMSE = \sqrt{\frac{1}{N}||\mathbf{x} - \hat{\mathbf{x}}||_{\ell 2}} \tag{9}$$

5.3 Datasets

Our experiments use a real dataset based in Iowa, US provided by the Iowa State University [5]. The dataset consists of 3 feeders and 239 nodes. Measurements from 1120 customers are recorded but only data aggregated at the secondary feeder level is available due to privacy issues. Thus, each node on the average is still an aggregation of less than 5 customers. The data is of a low aggregation level. For each customer, one year of load consumption measurements from 2017 (365 days) is provided in 1-hour intervals. In addition, the full topology is provided. We divide the dataset into training, validation and testing subsets by ratio 6:2:2. That is, 219 days, 73 days, 73 days for training, validation and testing respectively. The time series of each customer is normalized to values between 0 and 1.

5.4 Results and Discussion

Table 4. Varying hops of neighbor effect on STGCN prediction accuracy in MAE

STGCN Blocks \ Hops	1	2	3
1	1.228	1.216	1.200
2	1.282	1.199	1.218
3	1.326	1.325	1.220

5.4.1 STLF with Complete Data. First, we performed experiments on the data that was not modified for missing data. For each of the baseline algorithms, we trained a prediction model for forecasting with the Iowa dataset. The goal of this experiment is to evaluate the ability of the proposed models to accurately predict into the short-term future (1, 3 or 6 hours).

Before we compare the performance of different algorithms, we needed to investigate the effect of varying i in the input graph for STGCNs g^i, which is defined in Section 4.3 as the graph converted from power grid topology with edges connecting all neighbors within reach of i hops. This investigation determines the best g^i to use for the remaining experiments. Results of the investigation is recorded in Table 4. The main observation we can draw from Table 4 is that increasing the degree of each node by connecting neighbors of 2-hops away or higher gives better accuracy by increasing information exchange between the nodes. We see the best number of GCN layers for using g^2 is 2, while that for using g^3 is 1. For g^2, having more neighbors increases the benefits from using GCN layers as there is more correlations to be found. For both g^2 and g^3, there is diminishing returns as number of GCN layers increase, as once information is propagated to nodes too far away, there is not as much correlation to be found. Based on this result, for the remaining experiments, we use g^2 as the graph for STGCN Multi-hop and use 2 layers of Spatial-Temporal Blocks.

Table 5 shows the results of real load prediction 1, 3, and 6 hours into the future for each baseline. We can see that STGCN performs the best out of all the baselines with respect to all the evaluation metrics. For example, for 3 hours look ahead prediction, it has a MAE of 4.43% lower than that of CNN, the second best performing algorithm. This shows that STGCN is able to extract useful information out of neighboring customers that allows it to outperform CNN. As the difference between STGCN and CNN is due to the graph convolution layers, these improvement can be attributed to graph convolution passing information along neighbors. Furthermore, using multi-hop neighbors reduces the MAE by 7.54% compared to CNN. This shows that there is correlation between nodes further than 1-hop, as using a graph including multi-hop neighbors further improved the accuracy. Similar trends can be observed for other prediction windows, and in general, higher accuracy can be achieved for shorter prediction window since it is an easier task than predicting values further into the future.

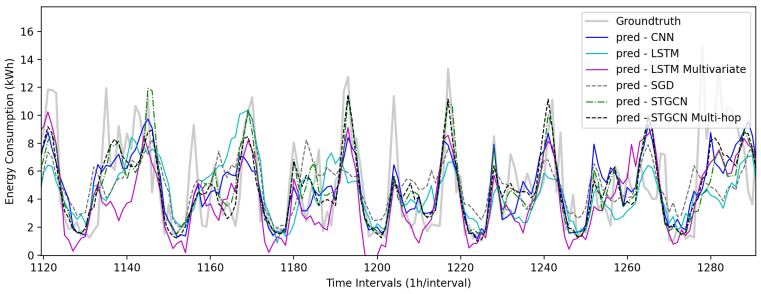

Fig. 3. Prediction results of one customer sample for real load

Figure 3 shows the 3 hours look ahead predictions for one customer sample over 7 days. The figure shows all models have difficulty predicting small volatile spikes, which is expected as such volatility is highly random and difficult to predict [13]. Relatively, STGCN is able to predict peaks in consumption more accurately.

Table 5. MAE, MAPE and RMSE for various algorithms for predicting real load with varying prediction window size

	MAE			MAPE (%)			RMSE		
Window (Hours)	1	3	6	1	3	6	1	3	6
SGD+Kernel	1.40369	1.41900	1.73481	21.391	25.564	30.520	3.20911	3.81044	5.35063
CNN	1.01794	1.28817	1.59557	21.169	24.564	33.913	2.22737	3.09290	4.15966
LSTM	1.05335	1.62771	1.73779	20.252	26.355	29.390	2.43127	4.08062	4.92000
LSTM Multivariate	1.12519	1.49880	1.94169	20.447	28.050	30.211	2.77536	3.94451	5.37146
STGCN	0.95911	1.23098	1.39626	18.970	23.178	25.431	2.00703	3.06287	4.30053
STGCN Multi-hop	**0.93002**	**1.19778**	**1.34846**	**18.899**	**23.059**	**25.086**	**1.91827**	**2.81910**	**3.69867**

5.4.2 STLF with Missing Data. For STLF with missing data experiments, we use a prediction window of 3 hours for testing the models. STGCN and STGCN Multi-hop is compared against the second best performing model, CNN, on the dataset used. This comparison also emphasizes the advantage provided by leveraging spatial information as STGCN and CNN mostly differ structurally with respect to the inclusion of graph convolution layers in STGCNs. The goal of this experiment is to evaluate the robustness of STGCN under a dataset with anomalies.

First half of Table 6 shows the results for Missing Data Experiment 1 and 2, where customers with missing data are chosen randomly, missing values are set as zeros or filled with linearly interpolated data respectively. In both experiments, both STGCN methods perform better than CNN for all the metrics and under all lengths of missing periods. STGCN Multi-hop performs better than STGCN in most cases. In general, we can observe that the improvements using STGCN increases as the length of missing period increases. For zero values, STGCN Multi-hop has a RMSE 26.26% lower than CNN when the length of missing period is 6 hours, and 38.72% lower when that length is 24 hours. For period of length generated by a real-life statistic distribution, STGCN Multi-hop still performs as well, having a RMSE 24.80% lower than that of CNN. Similar results can be observed for interpolated values, where STGCN Multi-hop has a RMSE 8.01% lower for 6 hours periods of missing data than CNN, 7.95% lower for 24 hours periods, and 5.52% for statistical distribution. This can also be observed in Figure 4 which plot a subsequence of the full testing prediction period of Missing Data Experiment 1 and 2 respectively. In both plots, a subset of time intervals is selected such that several

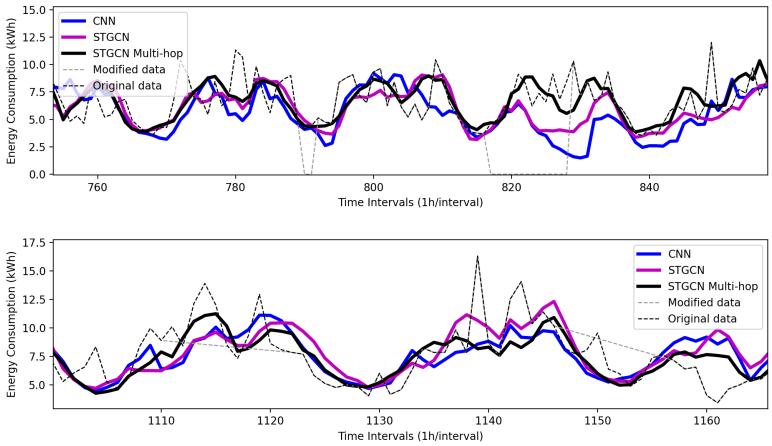

Fig. 4. Prediction results of Missing Data experiment 1 (Top) and 2 (Bottom)

periods of missing data can be observed. These time intervals are between intervals 780 to 800 and around interval 820 for Experiment 1 and between intervals 1100 to 1130 and 1140 to 1160 for Experiment 2. It can be seen clearly that STGCN prediction follows the original data more closely than CNN, and this is more noticeable during a missing data period. The reason behind this is that when data is missing for longer periods, the accuracy of CNN suffers as it relies only on temporal correlations whereas STGCNs are able to leverage data from neighboring customers. This shows that STGCN is more robust to missing data than state-of-the-art STLF models.

Additionally, STGCN achieves higher improvement over CNN when using dataset with missing values with respect to using data with no missing entries. Moreover, STGCN shows greater performance improvement when missing data is replaced with zero values than interpolated values. This is because zero values provide less information than interpolated values, and both cases of missing data provide less information than when there is no missing data. Thus, CNN performs relatively worse in the missing data cases as it has less information to use, while STGCN uses neighboring data to compensate. The experiments with fixed missing period lengths allow us to compare how performance changes with greater missing period lengths. The results show that in all cases, the error increases with longer missing periods. This can be explained by how the forecasting task is more difficult when data is missing in longer consecutive intervals. However, we can see that STGCN is less affected by longer missing values periods by utilizing neighbor information better.

Second half of Table 6 shows results of Missing Data Experiment 3 and 4 where missing data customers are selected by spatial locality and missing values are set as zeros or interpolated values. Similar trends are observed in these experiments as well, STGCN and STGCN Multi-hop performs better than CNN in all cases. The performance of STGCN and STGCN Multi-hop are similar. For Missing Data Experiment 3, in terms of RMSE, STGCN Multi-hop has a RMSE 6.56% lower for 6 hours periods of missing data than CNN, 16.69% lower for 24 hours periods, and 17.82% for statistical distribution. For Missing Data Experiment 4, in terms of RMSE, STGCN Multi-hop has a RMSE 6.92% lower for 6 hours periods of missing data than CNN, 15.02% lower for 24 hours periods, and 5.01% for statistical distribution. Spatial locality selection produces clusters of customers with missing data, which may better represent missing data distribution

Table 6. MAE, MAPE and RMSE for Missing Data experiments at various missing period

		MAE				MAPE (%)				RMSE			
Setup No.	Method	6	12	24	Stat	6	12	24	Stat	6	12	24	Stat
1	CNN	1.609	1.992	2.283	1.768	60.17	86.58	103.16	51.75	2.750	3.315	3.719	2.784
1	STGCN	1.381	1.641	1.826	1.509	31.59	35.27	38.35	26.55	2.450	2.884	3.167	2.484
1	STGCN Multi-hop	1.232	1.379	1.496	1.405	29.17	31.19	33.10	25.96	2.178	2.465	2.681	2.231
2	CNN	1.345	1.383	1.439	1.043	27.02	27.74	29.27	25.74	2.145	2.229	2.348	1.882
2	STGCN	1.290	1.306	1.336	1.007	24.78	25.11	25.18	24.45	1.990	2.013	2.102	1.756
2	STGCN Multi-hop	1.257	1.281	1.329	0.999	24.25	24.65	24.98	24.42	1.986	2.046	2.175	1.783
3	CNN	1.023	1.266	1.450	1.036	52.46	69.37	79.82	44.44	1.899	2.245	2.521	1.856
3	STGCN	0.908	1.066	1.182	0.925	29.68	34.23	37.80	26.59	1.782	2.041	2.242	1.689
3	STGCN Multi-hop	0.890	1.041	1.157	0.885	29.65	34.04	37.39	26.25	1.705	1.952	2.160	1.575
4	CNN	1.201	1.238	1.327	0.952	22.98	23.94	26.94	28.44	1.988	2.057	2.263	1.677
4	STGCN	1.135	1.152	1.194	0.909	20.68	20.98	21.26	26.43	1.844	1.866	1.993	1.574
4	STGCN Multi-hop	1.147	1.158	1.191	0.921	20.48	20.65	20.79	26.63	1.859	1.879	1.967	1.590

in real life cases than the previous two experiments. Thus, this experiment shows that STGCN is a very robust model for STLF.

6 CONCLUSION

In this paper, we investigated the benefits of leveraging spatial information in smart grids for the short term load forecasting (STLF) problem. Spatial-temporal Graph Convolutional Network (STGCN), a state-of-the-art neural network model that has shown success in other spatial-temporal data analysis problems, was used to introduce spatial information (or topological information) into the load consumption prediction model.

We compared the performance of STGCNs to state-of-the-art models like Convolutional Neural Networks (CNN) and Long Short-term Memory Recurrent Neural Networks (LSTM) for STLF. The performance was evaluated by calculating the error incurred in prediction consumption values 1, 3, and 6 hours in future for real loads. We also tested the robustness of each model by using a dataset which was modified to include missing values according to widely occurring missing data patterns in real power grids.

Our results showed that STGCNs outperformed all other baselines in real load prediction using a real life dataset based in Iowa, US. In particular, STGCN using a graph including 2-hop neighbors had a prediction Mean Absolute Error (MAE) of 7.54% lower than the second best performing model CNN. This demonstrates that there exists spatial correlations in customers load consumption data that could be exploited for more accurate load forecasting. It was also shown that STGCN performed better when edges connected 2-hop neighbors than simply using the original graph, suggesting that there are correlation between nodes that are further than 1-hop away. It was also shown that STGCN was highly robust to missing data compared to CNN, where the improvement in accuracy was as high as a reduction in RMSE by 38.72%. This is important because missing values in datasets is a common occurrence due to reasons like communication error in the grid.

ACKNOWLEDGMENTS

This work has been sponsored by the U.S. Army Research Office (ARO) under award number W911NF1910362 and the U.S. National Science Foundation (NSF) under award numbers 1911229 and 2009057.

REFERENCES

[1] Mohamed H Albadi and Ehab F El-Saadany. 2008. A summary of demand response in electricity markets. *Electric power systems research* 78, 11 (2008), 1989–1996.

[2] Hesham K Alfares and Mohammad Nazeeruddin. 2002. Electric load forecasting: literature survey and classification of methods. *International journal of systems science* 33, 1 (2002), 23–34.

[3] Syed Saqib Ali and Bong Jun Choi. 2020. State-of-the-Art Artificial Intelligence Techniques for Distributed Smart Grids: A Review. *Electronics* 9, 6 (2020), 1030.

[4] Mubbashra Anwar, Afrah Naeem, Hira Gul, Arooj Arif, Sahiba Fareed, and Nadeem Javaid. 2020. Electricity Price and Load Forecasting Using Data Analytics in Smart Grid: A Survey. In *International Conference on Emerging Internetworking, Data & Web Technologies*. Springer, 427–439.

[5] Fankun Bu, Yuxuan Yuan, Zhaoyu Wang, Kaveh Dehghanpour, and Anne Kimber. 2019. A time-series distribution test system based on real utility data. In *2019 North American Power Symposium (NAPS)*. IEEE, 1–6.

[6] Ervin Ceperic, Vladimir Ceperic, and Adrijan Baric. 2013. A strategy for short-term load forecasting by support vector regression machines. *IEEE Transactions on Power Systems* 28, 4 (2013), 4356–4364.

[7] Yongbao Chen, Peng Xu, Yiyi Chu, Weilin Li, Yuntao Wu, Lizhou Ni, Yi Bao, and Kun Wang. 2017. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Applied Energy* 195 (2017), 659–670.

[8] Xishuang Dong, Lijun Qian, and Lei Huang. 2017. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 119–125.

[9] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[10] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. 2017. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid* 10, 1 (2017), 841–851.

[11] Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2018. NO-LESS: Near optimal curtailment strategy selection for net load balancing in micro grids. In *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 1–5.

[12] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. 2016. Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks* 6 (2016), 91–99.

[13] Yayu Peng, Yishen Wang, Xiao Lu, Haifeng Li, Di Shi, Zhiwei Wang, and Jie Li. 2019. Short-term load forecasting at different aggregation levels with predictability analysis. In *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE, 3385–3390.

[14] Seunghyoung Ryu, Minsoo Kim, and Hongseok Kim. 2020. Denoising Autoencoder-Based Missing Value Imputation for Smart Meters. *IEEE Access* 8 (2020), 40656–40666.

[15] Sarabjit Singh and Rupinderjit Singh. 2015. ARIMA based short term load forecasting for Punjab region. *International Journal of Science and Research* 4, 6 (2015), p1819–1822.

[16] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. 2018. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid* 10, 3 (2018), 3125–3148.

[17] Christopher KI Williams and Matthias Seeger. 2001. Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*. 682–688.

[18] Xishuang Dong, Lijun Qian, and Lei Huang. 2017. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 119–125.

[19] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[20] Kasım Zor, Oğuzhan Timur, and Ahmet Teke. 2017. A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. In *2017 6th International Youth Conference on Energy (IYCE)*. IEEE, 1–7.